Combination of evidence for effective web search

Dong Nguyen
Carnegie Mellon University
Language Technologies Institute
Pittsburgh, PA 15213
dongn@cs.cmu.edu

Jamie Callan
Carnegie Mellon University
Language Technologies Institute
Pittsburgh, PA 15213
callan@cmu.edu

ABSTRACT

In this paper we describe Carnegie Mellon University's submission to the TREC 2010 Web Track. Our baseline run combines different methods, of which in particular the spam prior and mixture model were found the most effective. We also experimented with expansion over the Wikipedia corpus and found that picking the right Wikipedia articles for expansion can improve performance substantially. Furthermore, we did preliminary experiments with combining expansion over the Wikipedia corpus with expansion over the top ranked web pages.

1. INTRODUCTION

Carnegie Mellon University participated in the Ad Hoc task of the TREC 2010 Web track. Our experiments were carried out on the English subset of ClueWeb09 (Category A). We focused in particular on the ad hoc task, but also submitted to the diversity task. Our aim was to improve P@10 and MAP, we therefore did not employ special methods to improve the diversity.

We first investigate the effectiveness of different methods that have shown to work well in the past: priors, mixture model and the dependency model. These methods are then combined to provide a strong baseline. Next we experiment with different pseudo relevance feedback strategies. We explore expansion using the Wikipedia corpus and perform preliminary experiments to combine this with expansion over the top retrieved web pages.

We first describe related work and our retrieval framework. We then outline our submitted runs and present and discuss the results. The conclusion summarizes our findings and provides suggestions for future work.

2. RELATED WORK

The Web can be viewed as a graph where pages are connected through links. Both the connections as well as the text around these links (anchor text) differentiate web search from traditional text search and have been exploited in web search. The connections have been used to compute authority scores such as PageRank [16] and HITS [10]. Eiron and McCurley [7] performed an analysis of anchor text for web search. They found that anchor text behaves much like real world queries. Furthermore, they found that the homogeneity of results improved when using anchor text. The documents returned tended to focus on the most common meaning of the query.

Priors have shown to be very effective for web search. Not only authority priors such as PageRank, but also other kind of priors have been investigated in the past. In particular, a prior giving deeper urls less probability has shown to be effective in entry page search [12], a common information need in web search, and in web search in general [9]. In the TREC 2009 Web Track, spam was a major issue. Experiments showed that applying a spam prior improved the performance of TREC 2009 Web Track's systems substantially [5].

Web queries are often short and ambiguous. Therefore query expansion can help to increase performance. External expansion on a cleaner (e.g. Wikipedia) or larger (explored by [6]) dataset has proven to be effective in the past. In the TREC 2009 Web Track different approaches to expand queries were explored. Specifically, approaches expanding the query using external sources such as Wikipedia (University of Glasgow [14], University of Amsterdam [8]) or commercial search engines (University of Waterloo [17]) were explored because the initial results can be very noisy.

3. WEB TRACK

3.1 Tasks

The goal of the Web Track is to explore and evaluate Web retrieval technologies [3]. TREC 2010 Web Track contained three tasks: ad hoc, diversity and spam filtering. The ad hoc task ranks systems according to their performance based on manual relevance assessments. For every query, a specific information need was specified. For example, for '*iron*' only pages about iron as an essential nutrient are considered as relevant for the ad hoc task. With the diversity task, the goal is to return a ranked list that provides a complete coverage of the query and avoids redundancy.

3.2 Dataset

The ClueWeb09 dataset contains about 1 billion web pages collected in January and February 2009. Systems can submit runs on category B (subset of 50 million documents) or category A (full dataset).

3.3 Evaluation

The ad hoc task is evaluated using expected reciprocal rank and standard measures such as P@10 and MAP. For the TREC 2009 Web Track, the MTC method was used to estimate the MAP. The diversity task is evaluated using measures such as intent aware ERR, α-nDCG and the novelty- and rank-biased precision (NRBP) measure.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
NOV 2010		**00-00-2010 to 00-00-2010**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Combination of evidence for effective web search	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Carnegie Mellon University,Language Technologies Institute,Pittsburgh,PA,15213	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

Presented at the Nineteenth Text REtrieval Conference (TREC 2010) held in Gaithersburg, Maryland on 16-19 November 2010. The conference was co-sponsored by the National Institute of Standards and Technology (NIST), the Defense Advanced Research Projects Agency (DARPA), and the Advanced Research and Development Activity (ARDA).

14. ABSTRACT

In this paper we describe Carnegie Mellon University's sub- mission to the TREC 2010 Web Track. Our baseline run combines di erent methods, of which in particular the spam prior and mixture model were found the most e ective. We also experimented with expansion over the Wikipedia cor- pus and found that picking the right Wikipedia articles for expansion can improve performance substantially. Further- more, we did preliminary experiments with combining ex- pansion over the Wikipedia corpus with expansion over the top ranked web pages.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**7**	

4. RETRIEVAL FRAMEWORK

This section outlines the different retrieval components that we explored. Our final submitted runs combine these methods and are presented in the next section. We use the Lemur Project's search engine [1]. A stopword list is applied and words are stemmed with the Krovetz stemmer.

The results presented in this section are the results on the training set (Web Track 2009) and calculated using the Trec Eval program. Although in the Web Track 2009 evaluation MTC was used (eMAP and eP@10), we found the regular MAP and P@10 to be more robust when evaluating runs that were not included in the original pool.

4.1 Priors

4.1.1 Computing priors

We explore the effectiveness of three different priors: PageRank, Spam and an URL prior individually and combined with each other. For PageRank, we use the computed PageRank values provided by CMU [2]. For the Spam prior, we use the spam estimates made available by the University of Waterloo [5]. For every document, they provide an estimate of the percentile of documents in the corpus that are spammier than the particular document. The documents were mapped into two bins (percentile score less than or bigger than 50%) and log probabilities were computed for them. For the URL prior we followed the method described in Kamps et al. [9] (product squared variant).

4.1.2 Incorporating priors

Priors can be added in the Indri query language by adding the #PRIOR construct in the query. The following are two different variants for adding priors in the query. The first is the most straightforward way following the typical query likelihood model, by adding the priors directly in the query and treating it as a query term. In this way, the weight of the prior decreases when the query contains more terms.

```
#combine(#prior(PAGERANK) obama family tree)
```

When using multiple priors a composite prior is more suitable instead of adding all priors as query terms. Furthermore, a composite query can weight the composite prior and query terms. An example using two priors can be found below.

```
#weight(
   0.2 #weight(0.1 #prior(PAGERANK) 0.9 #prior(SPAM2))
   0.8 #combine(obama family tree )
)
```

4.1.3 Results

In Table 1 we compare the different priors individually. We included the prior using the first method, because this required no parameter tuning but gave us a good way to assess the effect of the different priors. In our final submitted runs we use the second method to include a composite prior.

Table 1 shows that the Spam prior is the most effective, both in Precision at 10 and MAP, which can be explained by the high amount of spam normally dominating the results. The URL prior is also effective, perhaps partially because spam pages are often very deep and short URL pages are less likely to be spam. Surprisingly, the PageRank prior alone does not perform very well. Comparing the results of the baseline with and without a PageRank prior, it seems that some popular, but not so relevant sites are promoted too much with a PageRank prior. Furthermore, some other relevant pages such as Wikipedia pages get a lower rank, probably because that particular Wikipedia article is not linked to often. However this only means PageRank is not effective as the only prior, combining it with the other priors can have additional benefits. For example, a (tuned) combination of PageRank and the Spam prior gave better performance than the Spam prior alone.

Table 1: Priors calculated over all relevance assessments of 2009.

Run	MAP	P@10
No prior	0.0647	0.1920
Spam	0.0745	0.2720
PageRank	0.0502	0.1820
Url	0.0657	0.2620

Table 2: Fields, calculated over all relevance assessments of 2009.

Run	MAP	P@10
Title	0.0203	0.0880
Title, spam	0.0219	0.1040
Inlink	0.0291	0.2300
Inlink, spam	0.0224	0.2440
Heading	0.0086	0.0600
Heading, spam	0.0121	0.0820
Document	0.0535	0.1180
Document, spam	0.0712	0.2340

4.2 Mixture model

Our index contains title, inlink, heading, and document fields. In Table 2 the results per field are shown. We follow the Indri query reference to construct a mixture model[1], an example query is:

```
#combine(
      #wsum( 5.0 espn.(title) 3.0 espn.(inlink) )
)
```

After tuning the parameters in combination with the priors, we observed the best performance with title (0.1), inlink (0.2) and document (0.7).

Inlink has a very high precision, which was also observed by others [11]. However, the MAP is low compared with using the document field, because not all sites have inlinks. Furthermore when adding a spam prior, the precision for the document field improves substantially, while the inlink precision improves much less. This again indicates that inlink is resistant to spam most of the times, while the document field is very susceptible to spam.

[1]http://www.lemurproject.org/lemur/IndriQueryLanguage.php

Table 3: Results dependency and mixture over all relevance assessments of 2009.

Run	MAP	P@10
Dependency model	0.0627	0.2560
Mixture model	0.0751	0.3120
Combination	0.0881	0.3360

4.3 Sequential dependency model

We employ the dependency model as proposed by Metzler and Croft [15]. This model expands the original query with subqueries that add proximity constraints. Both ordered and unordered constraints are added. After parameter tuning, we observed that the unordered component was not effective.

4.4 Combining dependency & mixture model

We combine the two models linearly by the #weight operator in Lemur and tune the weights using our training set.

```
#weight(
      0.9 #combine( mixture model )
      0.1 #combine( dependency model )
)
```

In Table 3 the effectiveness of the mixture model, dependency model and their combination is presented. The baseline system of these runs uses the optimized priors.

4.5 Host collapse

We also experimented with host collapse: only allowing a certain number of pages of the same hosts. The results however were mixed. Although this decreases the risk of certain pages appearing numerous times in the results (for example with slight variants due to dynamic generated pages), this also collapses useful websites such as Wikipedia.

4.6 Query expansion

We use the Relevance Model (RM1) by Lavrenko and Croft [13] to expand the queries.

4.6.1 Wikipedia expansion

Web pages are very noisy and even documents that are ranked highly in the results might not be suitable for query expansion. Therefore expansion over an external, cleaner corpus can give substantial performance benefits.

Advantages
Expansion over a Wikipedia corpus has several advantages over expansion using top ranked web pages. First, the corpus is very clean. Expanding over web pages has the risk of including noisy terms, such as spam terms or company names when searching for a commercial product (e.g. '*cheap internet*'). Second, Wikipedia especially has a good coverage of named entities. Third, Wikipedia is objective. In contrary to expansion over the web, which has a risk of drifting to a particular viewpoint if the top-ranked pages are very biased (which we observed for example in the query '*Rick Warren*').

Disadvantages
However, expanding over Wikipedia also brings some disadvantages. Queries such as '*getting organized*', '*cheap internet*', or '*travel information*', are not well covered by Wikipedia. Furthermore, Wikipedia articles do not always represent what the average Web user might be interested in. For example, a user issuing the query Yahoo might be interested in its services (such as '*search*', '*ask*', '*job*' etc.), while expanding over Wikipedia might add terms related to the company itself, such as acquisition and its development over the years. And lastly, expanding over Wikipedia often adds Wikipedia specific terms such as '*article*', '*edit*' and '*Wikipedia*' and thus biasing the results to Wikipedia pages which is sometimes not desirable.

Picking Wikipedia articles for expansion
To overcome the noise that Wikipedia might bring in with queries for which Wikipedia is not suitable, we explored the following strategy. We only consider Wikipedia pages for expansion that appear in the top results (e.g. top 1000) of the main search. This reduces the risk of expanding over Wikipedia when Wikipedia pages do not match the query well. Furthermore, the number of documents to estimate the relevance model with is not fixed and depends automatically on the match between the Wikipedia articles and the query (although we do set a maximum number of pages to be included).

This strategy is compared with the baseline strategy, which searches in the Wikipedia corpus directly. The advantage the baseline brings is that it is possible to optimize the queries to search in the Wikipedia corpus. For example, we observed that the title extent is more effective in the Wikipedia corpus than in general web search.

Exact match
We also explore the strategy which we call 'exact match'. If the title of a Wikipedia article matches the query exactly and is not a disambiguation page, we only use that page to expand. This happens often with named entity queries (such as '*the secret garden*' or '*starbucks*'). However, when the query is ambiguous (such as '*kcs*') no exact match is found and we use multiple pages. This strategy is often very accurate, because pages match only exactly when there is no ambiguity question or when there is a clear majority sense (for example for the query '*euclid*').

Summary variations
We explored the following strategies:

Expansion collection: Only use Wikipedia articles that have been returned in the top X results, or always expand a query by searching in the whole Wikipedia corpus.

Exact match: If the title of a Wikipedia article matches the query exactly, only use this page for expansion.

Results
Results are presented in Table 4. We observe that both proposed strategies, 'exact match' and only using the Wikipedia articles retrieved in the top documents improve both MAP as well as P@10 significantly.

Table 4: Wikipedia expansion over all relevance assessments of 2009.

Run	MAP	P@10
Only top documents, exact match	0.1399	0.4520
Top documents, no exact match	0.1169	0.3980
Whole wiki corpus, no exact match	0.1088	0.3540

4.6.2 Combining Wikipedia and expansion over top retrieved Web pages

In this section we explore combining Wikipedia expansion with expansion over the top retrieved web pages. Combining relevance feedback over different document samples has been investigated by Collins-Thompson and Callan [4]. However, our document samples are not random, and are very different in nature and quality from each other. We explore the following strategies:

Union: Expand by taking the union of the top terms of both expansion term sets.

Intersection: Expand by taking the intersection for both sets and averaging the weights.

Intersection
Using only terms that appear in the intersection yields the following advantages. If there is a mismatch between the pages returned by Wikipedia and Web expansion, (almost) no terms will be added. This thus decreases the risk of adding non-suitable terms. If there is a good match between two sets, more expansion terms are added. Furthermore, it automatically deletes terms that are noise (for example Wikipedia specific words such as '*edit*', '*article*' or noise from the Web expansion). However, the drawback is that if one of the expansion techniques performs poorly, (almost) no terms are added. Therefore difficult queries such as '*the music man*' or '*kcs*' are almost not expanded.

Union
This approach takes the top terms from every query expansion. Weights are not modified, except if a term appears in both sets, the weight of that term is the sum of the weights in each set. This approach introduces more noise, but guarantees that every query will be expanded. Furthermore, terms are added even if a query expansion does not give good expansion terms.

Examples
In Table 5 two examples are presented of a (non optimized) run which takes the intersection. We see that for these queries the intersection removes almost all the noise. Note that the terms that appear in the intersection are not necessarily the terms that have the highest weight in the Wikipedia or web expansion.

4.6.3 Incorporating the query terms in the query

The expansion terms were added in our baseline model (as described in the next section) by weighting the original terms and the new expansion terms using the #weight operator. In preliminary experiments we experimented with weights 0.3, 0.5 and 0.7. The observed differences were small, therefore we continued with using 0.5.

5. AD HOC: RESULTS & DISCUSSION

Parameters were tuned using parameter sweeps with the data from Web Track 2009. For the computationally heavy parameter sweeps, we used a subset of 20 queries. We will discus results on the submitted runs for the Web Track of TREC 2009 and 2010. Results for 2010 are for 48 of the 50 topics, topics 95 and 100 have been dropped.

5.1 Submission I: Baseline

5.1.1 Run description

Our final run makes use of the priors and combines the mixture and dependency model linearly. Note that this run does not contain PageRank and no unordered window with the dependency model. An example query of the baseline run is:

```
#weight(
   0.2 #weight(
      0.9 #prior(SPAM2)
      0.1 #prior(URL1))
   0.8 #weight(
      0.9 #combine(
         #wsum(
            0.1 milwaukee.(title) 0.2 milwaukee.(inlink)
            0.7 milwaukee.(document)
         ) #wsum(
            0.1 journal.(title) 0.2 journal.(inlink)
            0.7 journal.(document)
         ) #wsum(
            0.1 sentinel.(title) 0.2 sentinel.(inlink)
            0.7 sentinel.(document)
         )
       )  0.1 #weight (
            0.2 #combine(milwaukee journal sentinel)
            0.8 #combine (
               #1( milwaukee journal)
               #1( journal sentinel)
               #1( milwaukee journal sentinel)
            )
       )
   )
)
```

5.1.2 Discussion

Analyzing the queries for which our system performed poorly, we can make three observations. First, stop word removal definitely made some queries harder with the most extreme example the 2010 query '*to be or not to be that is the question*' (only retaining '*question*'). But even for more common queries stopword removal can make a big difference (for example '*music man*' versus '*the music man*' or '*wall*' versus '*the wall*'). Second, stemming also decreased the performance heavily in some queries. For example for the 2010 query '*living in india*', '*living*' matched '*live*' resulting in many results about live webcams, live sport results etc. A 2009 query example is '*the current*' (with stemming '*currency*' matched to the query). The last type of queries for which the performance was low was ambiguous queries, such as '*defender*' or '*kcs*' or for 2010 the queries '*avp*' or '*iron*'. We found that 30 of the 48 queries were on or above median regarding P@10. For 2 queries it had the best P@10.

Table 5: Expansion terms.

Query	Wikipedia expansion	Web expansion	Intersection
orange county convention center	center, convention, county, phase, orange, 2, space, ft, m, orlando, sq, wikipedia, million, build, 1, tourist, north, exhibition, florida, new, state, complete, approve, occc, tax, bcc, south, drive, january, 000	center, convention, orange, county, occc, hotel, feb, show, event, md, san, diego, international, rental, www, 1, com, new, service, florida, 4, 407, near, net, 5, 00, home, restaurant, orlando, drive	convention, drive, new, orange, florida, orlando, 1, county, occc, center
dogs adoption	animal, dog, greyhound, pet, adoption, wikipedia, race, shelter, group, article, home, page, edit, adopt, own, org, wiki, http, care, en, rescue, link, 3, state, need, category, free, work, marine, live	dog, adoption, cat, shelter, rescue, pet, event, animal, 1, adopt, ny, north, new, hempstead, home, 3, near, york, washington, 4, 2, port, breed, month, information, puppy, care, service, link, com	home, adopt, shelter, dog adoption, pet, rescue link, animal, care, 3

Table 6: Ad hoc: MAP and P@10.

	2009		2010	
Run	**MAP**	**P@10**	**MAP**	**P@10**
Baseline (cmuBase10)	0.0881	0.3360	0.0976	0.2833
Wikipedia (cmuWiki10)	0.1399	0.4520	0.1574	0.4208
Union (cmuFuTop10)	0.1040	0.3320	0.1177	0.3125
Int.+add. (cmuComb10)	0.1137	0.3780	0.1209	0.3250

Table 7: Ad hoc: ERR@20 and nDCG@20.

	2010	
Run	**ERR@20**	**nDCG@20**
Baseline (cmuBase10)	0.09130	0.14200
Wikipedia (cmuWiki10)	0.11206	0.21181
Union (cmuFuTop10)	0.10009	0.15825
Inters.+add. (cmuComb10)	0.09831	0.16899

5.2 Submission II: Wikipedia expansion

5.2.1 Run description

Our second submission only uses Wikipedia for query expansion. Only Wikipedia pages appearing in the top results (e.g. top 1000) are considered for expansion. Furthermore, we apply the 'exact match' strategy. We take the top 10 Wikipedia articles, extract 30 expansion terms and give the expansion query a weight of 0.5. We furthermore applied a stopword list with Wikipedia specific terms (such as '*edit*'). Expansion terms are integrated in our baseline system.

5.2.2 Discussion

Compared to the baseline both MAP and P@10 increases a lot with this run. Examples for which the P@10 increased heavily are '*joints*' (+0.8), '*korean language*' (+1) and '*raffles*' (+0.9). '*joints*' and '*korean language*' are two concepts both well represented by Wikipedia and the 'exact match' strategy was applied in these cases. '*raffles*' is ambiguous, therefore multiple Wikipedia articles were used for expansion and terms such as '*hotel*','*travel*' and '*singapore*' were added, matching the right intent ('*Find the homepage of Raffles Hotel in Singapore*'). However, for some queries the performance also decreased. An example is '*discovery channel store*' (-0.8), a typical example of queries for which the encyclopedic character of Wikipedia might not be suitable. 33 of the 48 queries were on or above median regarding P@10. For 10 queries it had the best P@10.

5.3 Submission III: Combining Wikipedia and Web expansion

Our third submission combines Wikipedia and expansion over the top retrieved web pages and uses the union approach. For our diversity track, we also submitted a run with the intersection approach and additional modifications (cmuComb10, see next section). However, because this run was also evaluated with the ad hoc measures, these are also reported in Table 6 and 7. For Wikipedia expansion, we used the exact match strategy but searched in the Wikipedia corpus directly. We used the top 10 documents for both expansion methods and included the top 10 expansion terms of each method. We applied host collapse before doing expansion on the web documents. Due to time constraints, parameters for this run were not tuned.

5.3.1 Discussion

Compared to only Wikipedia expansion, the performance is worse. A more effective approach could be to give Wikipedia terms more weight, or depending on the match of Wikipedia or web articles with the query to expand only using one method. Furthermore, simply leaving the weights unchanged might not be the most effective method. However, we did observe that some queries performed better than the baseline or Wikipedia expansion. For example '*sewing instructions*' got a P@10 of 0.9 compared to 0.6 (baseline) and 0.1 (Wikipedia expansion). For this query, expansion over the web documents (adding '*manual*', '*book*', '*machine*', '*pattern*' etc.) gave more suitable terms than expansion over Wikipedia (adding '*home*', '*company*', '*style*', '*gar*' etc.). On average, we found that the performance tended to be closer to the baseline than to Wikipedia expansion. 31 of 48 of the queries were on or above median regarding P@10. For 3 queries it had the best P@10.

For the intersection approach, the performance is also lower compared to Wikipedia expansion. Direct comparison with the union approach is not possible, because this run contains additional modifications. 35 of 48 of the queries were on or above median regarding P@10. For 4 queries it had the best P@10. This run has the highest number of queries on or above median, although the P@10 is much lower than the Wikipedia expansion run. This indicates that the Wikipedia run had a more extreme behavior (performing very well or very bad on some queries), which can be explained by the expansion approach.

Table 8: Diversity results for submitted runs (+ baseline) 2010.

Run	ERR-IA@20	α-nDCG@20	NRBP	MAP-IA
Baseline (cmuBase10)	0.201887	0.304172	0.163077	0.049794
Wikipedia (cmuWiki10)	0.248370	0.345176	0.214939	0.092600
Union (cmuFuTop10)	0.208400	0.309083	0.170817	0.062064
Inters.+add. (cmuComb10)	0.215057	0.323582	0.173160	0.064907

6. DIVERSITY: RESULTS & DISCUSSION

6.1 Runs submitted

We submitted two runs that were also submitted to the ad hoc task: Wikipedia expansion and the combination of Wikipedia and web expansion. Our third run was also a combination of Wikipedia and Web expansion, but using the intersection method. Furthermore, for the third run we experimented with adding the expansion terms with a mixture model. In addition, when the query matched exactly with an Wikipedia article and the query contained articles (such as '*the*'), we added all the expansion terms obtained by expansion over the Wikipedia corpus.

6.2 Discussion

Results of the 2010 runs can be found in Table 8. Unfortunately, it is difficult to draw hard conclusions, because the diversity of the queries also depends on the precision of the results. The Wikipedia run is the best according to all measures. However, this might be because Wikipedia has the highest precision. For some of the queries, adding expansion terms from both web and Wikipedia expansion improved diversity compared to the baseline. For example for the query '*titan*', the P@10 for the union approach is 0.3, the same as for the baseline. However, the α-nDCG@10 increases from 0.172 to 0.409. Added terms were: *titan, nissan, internet, poker, review, luggage, cab, pickup, bed, tennessee, titan, teen, team, series, new, member, vol, comic, issue* and *2008*, covering a wide range of different topics.

7. CONCLUSION

In this paper we described our participation in the TREC 2010 Web Track. We first explored several approaches that worked well in the past, such as priors, mixture model and dependency model. We found the most effective prior to be the spam prior, and the mixture model was more effective than the dependency model. The best baseline model combined all three components.

We observed that expansion over the Wikipedia corpus is very effective, dramatically increasing the performance over some queries for which the initial results were very poor. Furthermore, selecting Wikipedia articles for expansion is more effective when the documents that are picked appear high in the main search, instead of directly searching in the Wikipedia corpus. We also explored an approach that combines the query expansion over different collections (top results and Wikipedia). However, performance on average was lower than only using Wikipedia expansion, especially on queries for which the initial results were very poor.

For future work, we expect that a more sophisticated approach to combine Wikipedia expansion with expansion over the top documents can be more effective. For example, only relying on one expansion method when the expansion collection seem to match the query well, or giving a particular expansion method more weight.

8. ACKNOWLEDGMENTS

We would like to thank Liu Liu and Minh Duong for providing the spam priors in the Lemur format. This paper is based on research funded by NSF grant NSF EEC 0935127.

9. REFERENCES

[1] The lemur project - http://www.lemurproject.org/.

[2] PageRank ClueWeb09 provided by CMU - http://boston.lti.cs.cmu.edu/clueweb09/wiki/tiki-index.php?page=pagerank.

[3] C. L. A. Clarke, N. Craswell, and I. Soboroff. Overview of the trec 2009 web track. In *Proceedings of the Eighteenth Text REtrieval Conference*, 2010.

[4] K. Collins-Thompson and J. Callan. Estimation and use of uncertainty in pseudo-relevance feedback. In *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 303–310, New York, NY, USA, 2007. ACM.

[5] G. V. Cormack, M. D. Smucker, and C. L. A. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *CoRR*, abs/1004.5168, 2010.

[6] F. Diaz and D. Metzler. Improving the estimation of relevance models using large external corpora. In *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 154–161, New York, NY, USA, 2006. ACM.

[7] N. Eiron and K. S. McCurley. Analysis of anchor text for web search. In *SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 459–460, New York, NY, USA, 2003. ACM.

[8] J. He, K. Balog, K. Hofmann, E. J. Meij, M. de Rijke, E. Tsagkias, and W. Weerkamp. Heuristic ranking and diversification of web documents. In *Proceedings of the Eighteenth Text REtrieval Conference*, February 2010.

[9] J. Kamps, G. Mishne, and M. D. Rijke. Language models for searching in web corpora. In *Proceedings of the Thirteenth Text REtrieval Conference*, 2004.

[10] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[11] M. Koolen and J. Kamps. The importance of anchor text for ad hoc search revisited. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, pages 122–129, 2010.

[12] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 27–34, New York, NY, USA, 2002. ACM.

[13] V. Lavrenko and W. B. Croft. Relevance based language models. In *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 120–127, New York, NY, USA, 2001. ACM.

[14] R. McCreadie, M. Craig, O. Iadh, P. Jie, and L. T. S. Rodrygo. University of Glasgow at TREC 2009: Experiments with Terrier – Blog, Entity, Million Query, Relevance Feedback, and Web tracks. In *Proceedings of the Eighteenth Text REtrieval Conference*, 2010.

[15] D. Metzler and W. B. Croft. A Markov random field model for term dependencies. In *SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 472–479, New York, NY, USA, 2005. ACM.

[16] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[17] M. D. Smucker, C. L. A. Clarke, and G. V. Cormack. Experiments with clueweb09: Relevance feedback and web tracks. In *Proceedings of the Eighteenth Text REtrieval Conference*, February 2010.